MPLS Traffic Engineering: Interarea Tunnels

First Published: January 16, 2003
Last Updated: January 4, 2007

The MPLS Traffic Engineering: Interarea Tunnels feature allows you to establish Multiprotocol Label Switching (MPLS) traffic engineering (TE) tunnels that span multiple Interior Gateway Protocol (IGP) areas and levels, removing the restriction that had required the tunnel headend and tailend routers both be in the same area. The IGP can be either Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF).

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature History for MPLS Traffic Engineering: Interarea Tunnels" section on page 24.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for MPLS Traffic Engineering: Interarea Tunnels

Your network must support the following Cisco IOS features:

- MPLS
- IP Cisco Express Forwarding
- IS-IS or OSPF
- Traffic engineering tunnels

Restrictions for MPLS Traffic Engineering: Interarea Tunnels

- The dynamic path option feature for TE tunnels (the **tunnel mpls traffic-eng path-option** *number* **dynamic** command) is not supported for interarea tunnels. An explicit path identifying the Area Border Routers (ABRs) is required. When there are choices for the ABRs to be used, multiple explicit paths are recommended, each of which identifies a different sequence of ABRs.
- The MPLS TE AutoRoute feature (the **tunnel mpls traffic-eng autoroute announce** command) is not supported for interarea tunnels because you would need to know the network topology behind the tailend router.
- Tunnel affinity (the **tunnel mpls traffic-eng affinity** command) is not supported for interarea tunnels.
- The reoptimization of tunnel paths is not supported for interarea tunnels.

Information About MPLS Traffic Engineering: Interarea Tunnels

Before using the MPLS Traffic Engineering: Interarea Tunnels feature, you need to understand the following concepts:

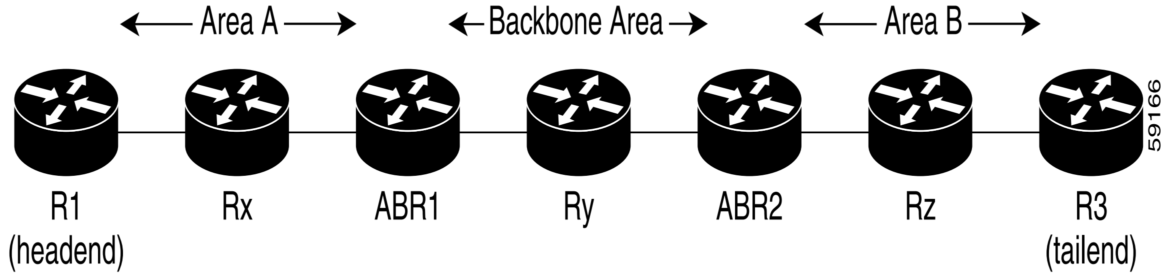
- Interarea Tunnels Functionality, page 2
- Interarea Tunnels Benefits, page 3

Interarea Tunnels Functionality

To configure an interarea tunnel, you specify on the headend router a loosely routed explicit path for the tunnel label-switched path (LSP) that identifies each ABR the LSP should traverse using the **next-address loose** command. The headend router and the ABRs along the specified explicit path expand the loose hops, each computing the path segment to the next ABR or tunnel destination.

For example, to configure a TE tunnel from router R1 to router R2 in the simple multiarea network shown in Figure 1, you would specify ABR1 and ABR2 as loose hops in the explicit path for the tunnel. To signal the tunnel LSP, the headend router (R1) computes the path to ABR1 and sends a Resource Reservation Protocol (RSVP) Path message specifying the path from itself to ABR1 as a sequence of strict hops followed by the path from ABR1 to the tailend as a sequence of loose hops (ABR2, R1). When ABR1 receives the Path message, it expands the path across the backbone area to ABR2 and forwards the Path message specifying the path from itself to ABR2 as a sequence of strict hops followed by the path from ABR2 to the tunnel tailend (R3) as a loose hop. When ABR2 receives the Path message, it expands the path across the tailend area to R3 and propagates the Path message specifying the path from itself to R2 as a sequence of strict hops.

Figure 1 *Multiarea Network*



Note Cisco IOS Release 12.2(33)SRB supports stateful switchover (SSO) recovery of LSPs that include loose hops.

Note Strictly speaking, IS-IS does not have the notion of an ABR. For the purpose of discussing the MPLS Traffic Engineering: Interarea Tunnels feature, this document considers an IS-IS level-1-2 router as an ABR.

Note The explicit path for a TE interarea tunnel may contain any number of non-ABR LSRs. Within an area, a combination of loose and strict next IP addresses is allowed. To specify the next IP address in the explicit path, use the **next-address** command.

Note With OSPF, if an area is connected to the backbone through a virtual link, there may be more than two ABRs in the path.

The following MPLS TE features are supported on interarea traffic engineering LSPs:

- Automatic bandwidth adjustment
- Diff-Serve-aware traffic engineering
- Fast reroute link protection
- Policy-based routing
- Static routing

Interarea Tunnels Benefits

When it is desirable for the traffic from one router to another router in a different IGP area to travel over TE LSPs, the MPLS Traffic Engineering: Interarea Tunnels feature allows you to configure a tunnel that runs from the source router to the destination router. The alternative would be to configure a sequence of tunnels, each crossing one of the areas between source and destination routers such that the traffic arriving on one such tunnel is forwarded into the next such tunnel.

How to Configure MPLS Traffic Engineering: Interarea Tunnels

This section contains the following tasks:

- Configuring OSPF for Interarea Tunnels, page 4 (optional)
- Configuring IS-IS for Interarea Tunnels, page 7 (optional)
- Configuring an MPLS Traffic Engineering Interarea Tunnel, page 11 (required)
- Verifying the MPLS Traffic Engineering: Interarea Tunnels Configuration, page 14 (optional)

Note You must configure either OSPF or IS-IS.

Configuring OSPF for Interarea Tunnels

This section describes the following tasks:

- Configuring OSPF for ABR Routers, page 4
- Configuring OSPF for Non-ABR Routers, page 5

Configuring OSPF for ABR Routers

For each ABR that is running OSPF, perform the following steps to configure traffic engineering on each area you want tunnels in or across. By having multiple areas and configuring traffic engineering in and across each area, changes within the network can be contained within an area.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **network** ip-*address wildcard-mask* **area** *area-id*
5. **mpls traffic-eng router-id** *interface-name*
6. **mpls traffic-eng area 0**
7. **mpls traffic-eng area** *x*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 1`	Enables OSPF and enters router configuration mode. The *process-id* argument is an internally used identification parameter for the OSPF routing process. It is logically assigned and can be any positive integer. Assign a unique value for each OSPF routing process.
Step 4	**network** *ip-address wildcard-mask* **area** *area-id* **Example:** `Router(config-router)# network 192.168.45.0 0.0.255.255 area 1`	Specifies the interfaces on which OSPF is to run and specifies the area to which the interface is connected.
Step 5	**mpls traffic-eng router-id** *interface-name* **Example:** `Router(config-router)# mpls traffic-eng router-id Loopback0`	Specifies that the traffic engineering router identifier for the node is the IP address associated with a given interface. (The router identifier is displayed in the **show mpls traffic-eng topology path** command output.) **Note** The *interface-name* must be Loopback0.
Step 6	**mpls traffic-eng area 0** **Example:** `Router(config-router)# mpls traffic-eng area 0`	Turns on MPLS traffic engineering for OSPF in area 0.
Step 7	**mpls traffic-eng area** *x* **Example:** `Router(config-router)# mpls traffic-eng area 2`	Turns on MPLS traffic engineering for OSPF in area *x*.
Step 8	**end** **Example:** `Router(config-router)# end`	Returns to privileged EXEC mode.

Configuring OSPF for Non-ABR Routers

For each non-ABR that is running OSPF, perform the following steps.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router ospf** *process-id*
4. **network** *ip-address wildcard-mask* **area** *area-id*
5. **mpls traffic-eng router-id** *interface-name*
6. **mpls traffic-eng area** *x*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 1`	Enables OSPF and enters router configuration mode. The *process-id* argument is an internally used identification parameter for the OSPF routing process. It is locally assigned and can be any positive integer. Assign a unique value for each OSPF routing process.
Step 4	**network** *ip-address wildcard-mask* **area** *area-id* **Example:** `Router(config-router)# network 192.168.10.10 255.255.255.0 area 1`	Specifies the interfaces on which OSPF is to run and specifies the area to which the interface is connected.
Step 5	**mpls traffic-eng router-id** *interface-name* **Example:** `Router(config-router)# mpls traffic-eng router-id Loopback0`	Specifies that the traffic engineering router identifier for the node is the IP address associated with a given interface. (The router identifier is displayed in the **show mpls traffic-eng topology path** command output.) **Note** The *interface-name* must be Loopback0.
Step 6	**mpls traffic-eng area** *x* **Example:** `Router(config-router)# mpls traffic-eng area 1`	Specifies the area that the router is in.
Step 7	**end** **Example:** `Router(config-router)# end`	Returns to privileged EXEC mode.

Configuring IS-IS for Interarea Tunnels

This section describes the following tasks:

- Configuring IS-IS for Backbone Routers, page 7
- Configuring IS-IS for Nonbackbone Routers, page 8
- Configuring IS-IS for Interfaces, page 10

Configuring IS-IS for Backbone Routers

To configure IS-IS for background (level-1-2) routers, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **clns routing**
4. **router isis**
5. **metric-style wide**
6. **net** *nn.nnnn.nnnn.nnnn.nnnn*
7. **mpls traffic-eng router-id** *interface-name*
8. **mpls traffic-eng level-1**
9. **mpls traffic-eng level-2**
10. **interface** *typeslot/port*
11. **ip router isis**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`clns routing`** **Example:** `Router(config)# clns routing`	Allows IS-IS to communicate with other IS-IS-configured routers.

	Command or Action	Purpose
Step 4	**router isis** **Example:** `Router(config)# router isis`	Enables IS-IS routing and specifies an IS-IS process for IP. This command places the router in router configuration mode.
Step 5	**metric-style wide** **Example:** `Router(config-router)# metric-style wide`	Configures a router to generate and accept only new-style type, length, value objects (TLVs).
Step 6	**net** *nn.nnnn.nnnn.nnnn.nnnn* **Example:** `Router(config-router)# net 49 0000.0100.0000.0010`	Configures the area ID (area address) and the system ID.
Step 7	**mpls traffic-eng router-id** *interface-name* **Example:** `Router(config-router)# mpls traffic-eng router-id Loopback0`	Specifies that the traffic engineering router identifier for the node is the IP address associated with interface Loopback0.
Step 8	**mpls traffic-eng level-1** **Example:** `Router(config-router)# mpls traffic-eng level-1`	Turns on MPLS traffic engineering for IS-IS at level 1.
Step 9	**mpls traffic-eng level-2** **Example:** `Router(config-router)# mpls traffic-eng level-2`	Turns on MPLS traffic engineering for IS-IS at level 2.
Step 10	**interface** *typeslot***/***port* **Example:** `Router(config-router)# interface POS1/0`	Configures an interface type and enters interface configuration mode.
Step 11	**ip router isis** **Example:** `Router(config-if)# ip router isis`	Enables IS-IS routing. Specify this command on each interface on which you want to run IS-IS.
Step 12	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring IS-IS for Nonbackbone Routers

To configure IS-IS for nonbackbone routers, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **clns routing**
4. **router isis**
5. **metric-style wide**
6. **net** *nn.nnnn.nnnn.nnnn.nnnn*
7. **mpls traffic-eng router-id** *interface-name*
8. **mpls traffic-eng** {**level-1** | **level-2**}
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**clns routing** **Example:** `Router(config)# clns routing`	Allows IS-IS to communicate with other IS-IS-configured routers.
Step 4	**router isis** **Example:** `Router(config)# router isis`	Enables IS-IS routing and specifies an IS-IS process for IP. This command places the router in router configuration mode.
Step 5	**metric-style wide** **Example:** `Router(config-router)# metric-style wide`	Configures a router to generate and accept only new-style TLVs.
Step 6	**net** *nn.nnnn.nnnn.nnnn.nnnn* **Example:** `Router(config-router)# net 49.0000.2000.0100.0001`	Configures the area ID (area address) and the system ID.

	Command or Action	Purpose
Step 7	**mpls traffic-eng router-id** *interface-name* **Example:** `Router(config-router)# mpls traffic-eng router-id Loopback0`	Specifies that the traffic engineering router identifier for the node is the IP address associated with interface Loopback0.
Step 8	**mpls traffic-eng** {**level-1** | **level-2**} **Example:** `Router(config-router)# mpls traffic-eng level-1`	Turns on MPLS traffic engineering for IS-IS at level 1.
Step 9	**end** **Example:** `Router(config-router)# end`	Returns to privileged EXEC mode.

Configuring IS-IS for Interfaces

To configure IS-IS for interfaces, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **clns routing**
4. **router isis**
5. **metric-style wide**
6. **net** *nn.nnnn.nnnn.nnnn.nnnn*
7. **mpls traffic-eng router-id** *interface-name*
8. **interface** *typeslot/port*
9. **ip router isis**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**clns routing** **Example:** `Router(config)# clns routing`	Allows IS-IS to communicate with other IS-IS-configured routers.
Step 4	**router isis** **Example:** `Router(config)# router isis`	Enables IS-IS routing and specifies an IS-IS process for IP. This command places the router in router configuration mode.
Step 5	**metric-style wide** **Example:** `Router(config-router)# metric-style wide`	Configures a router to generate and accept only new-style TLVs.
Step 6	**net** *nn.nnnn.nnnn.nnnn.nnnn* **Example:** `Router(config-router)# net 49.0000.0100.0000.0010`	Configures the area ID (area address) and the system ID.
Step 7	**mpls traffic-eng router-id** *interface-name* **Example:** `Router(config-router)# mpls traffic-eng router-id Loopback0`	Specifies that the traffic engineering router identifier for the node is the IP address associated with interface Loopback0
Step 8	**interface** *typeslot***/***port* **Example:** `Router(config-router)# interface POS1/0`	Specifies the interface and enters interface configuration mode.
Step 9	**ip router isis** **Example:** `Router(config-if)# ip router isis`	Enables IS-IS routing. Specify this command on each interface on which you want to run IS-IS.
Step 10	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring an MPLS Traffic Engineering Interarea Tunnel

This section includes the following tasks:

- Configuring an MPLS Traffic Engineering Interarea Tunnel to Use Explicit Paths, page 12
- Configuring Explicit Paths, page 13

Configuring an MPLS Traffic Engineering Interarea Tunnel to Use Explicit Paths

To configure an MPLS traffic engineering interarea tunnel to use explicit paths, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *tunnel-interface*
4. **ip unnumbered** *type number*
5. **tunnel destination** *ip-address*
6. **tunnel mode mpls traffic-eng**
7. **tunnel mpls traffic-eng bandwidth** *bandwidth*
8. **tunnel mpls traffic-eng path-option** *number* **explicit** {**name** *path-name* | *path-number*} [**lockdown**]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *tunnel-interface* **Example:** `Router(config)# interface Tunel1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip unnumbered** *type number* **Example:** `Router(config-if)# ip unnumbered Loopback 0`	Gives the tunnel interface an IP address. An MPLS traffic engineering tunnel interface should be unnumbered because it represents a unidirectional link.
Step 5	**tunnel destination** *ip-address* **Example:** `Router(config-if)# tunnel destination 192.168.20.20`	Specifies the destination for a tunnel. You must enter the MPLS traffic engineering router ID of the destination device.

	Command or Action	Purpose
Step 6	**tunnel mode mpls traffic-eng** **Example:** `Router(config-if)# tunnel mode mpls traffic-eng`	Sets the tunnel encapsulation mode to MPLS traffic engineering.
Step 7	**tunnel mpls traffic-eng bandwidth** *bandwidth* **Example:** `Router(config-if)# tunnel mpls traffic-eng bandwidth 300`	Configures the bandwidth required for the MPLS traffic engineering tunnel.
Step 8	**tunnel mpls traffic-eng path-option** *number* **explicit** {**name** *path-name* | *path-number*} [**lockdown**] **Example:** `Router(config-if)# tunnel mpls traffic-eng path-option 1 explicit name path-Tunnel1`	Configures the tunnel to use a named IP explicit path or a path dynamically calculated from the traffic engineering topology database. The **path-option** keyword must specify the ABRs the tunnel LSP must traverse as loose hops via the **next-address loose** command.
Step 9	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring Explicit Paths

To configure explicit paths, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip explicit-path name** *pathname*
4. **next-address loose** *A.B.C.D.*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip explicit-path name** *pathname* **Example:** `Router(config)# ip explicit-path name path-tunnel1`	Enters the command mode for IP explicit paths and creates or modifies the specified path.
Step 4	**next-address loose** *A.B.C.D.* **Example:** `Router(configcfg-ip-expl-path)# next-address loose 192.168.40.40`	Specifies the next loose IP address in the explicit path. Each ABR the path must traverse should be specified in a **next-address loose** command.
Step 5	**end** **Example:** `Router(configcfg-ip-expl-path)# end`	Returns to privileged EXEC mode.

Verifying the MPLS Traffic Engineering: Interarea Tunnels Configuration

To verify that the MPLS TE topology database maintained by the ABR includes the topologies of each area to which it is connected for which MPLS TE is enabled, perform the following steps. Enter the **show mpls traffic-eng topology** command at each ABR in the network.

SUMMARY STEPS

1. **enable**
2. **show mpls traffic-eng topology**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show mpls traffic-eng topology** **Example:** `Router> show mpls traffic-eng topology`	Displays the MPLS traffic engineering global topology currently known at this node. Enter this command at each ABR in the network.

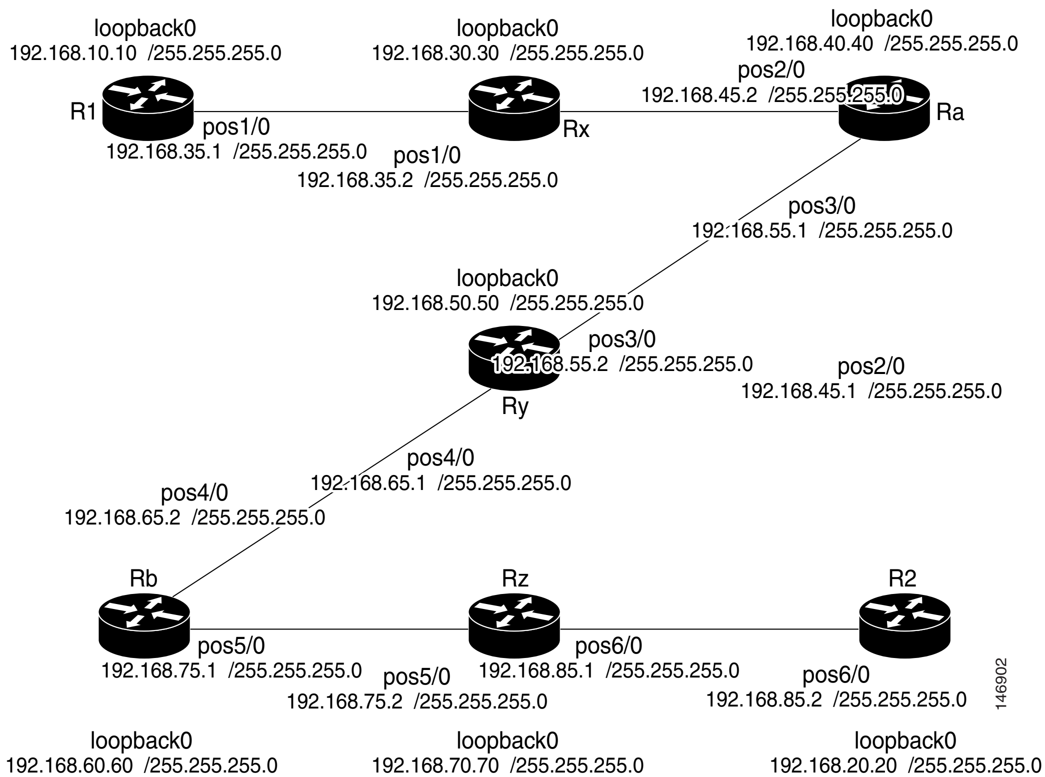
Configuration Examples for MPLS Traffic Engineering: Interarea Tunnels

This section shows how to configure MPLS traffic engineering interarea tunnels for the simple router topology illustrated in Figure 2. It includes configuration fragments that illustrate the configurations shown in the following sections:

- Configuring OSPF for Interarea Tunnels: Example, page 15
- Configuring IS-IS for Interarea Tunnels: Example, page 17
- Configuring MPLS and RSVP to Support Traffic Engineering: Example, page 18
- Configuring an Interarea Tunnel: Example, page 19

Note The examples concentrate on the configuration required to support interarea tunnels. Additional configuration for MPLS traffic engineering is possible. Refer to the *Cisco IOS IP Routing Protocols Configuration Guide,* Release 12.4.

Figure 2 Router Topology



Configuring OSPF for Interarea Tunnels: Example

The following configuration fragments show how to configure OSPF for interarea tunnels assuming that:

- Routers R1, Rx, and Ra are in OSPF Area 1

- Routers Ra, Ry, and Rb are in OSPF Area 0
- Routers Rb, Rz, and R2 are in OSPF Area 2
- Router Ra is an ABR for Area 0 and Area 1
- Router Rb is an ABR for Area 0 and Area 2

Router R1 OSPF Configuration

```
router ospf 1
 network 192.168.10.10 255.255.255.0 area 1
 network 192.168.35.0 0.0.255.255 area 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 1
```

Router Rx OSPF Configuration

```
router ospf 1
 network 192.168.30.30 area 1
 network 192.168.35.0 0.0.255.255 area 1
 network 192.168.45.0 0.0.255.255 area 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 1
```

Router Ra OSPF Configuration

Ra is an ABR for Area 0 and Area 1. Interface POS2/0 is in Area 1 and interface POS3/0 is in Area 0. The **mpls traffic-eng area** commands configure Ra for IGP TE updates for both areas.

```
router ospf 1
 network 192.168.40.40 area 0
 network 192.168.45.0 0.0.255.255 area 1
 network 192.168.55.0 0.0.255.255 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 mpls traffic-eng area 1
```

Router Rb OSPF Configuration

Rb is an ABR for Area 0 and Area 2. Interface POS4/0 is in Area 0 and interface POS5/0 is in Area 2. The **mpls traffic-eng area** commands configure Rb for IGP TE updates for both areas.

```
router ospf 1
 network 192.168.60.60 0.0.0.0 area 0
 network 192.168.65.0 0.0.255.255 area 0
 network 192.168.75.0 0.0.255.255 area 2
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 mpls traffic-eng area 2
```

Router Rz OSPF Configuration

```
router ospf 1
 network 192.168.70.70 0.0.0.0 area 2
 network 192.168.75.0 0.0.255.255 area 2
 network 192.168.85.0 0.0.255.255 area 2
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 2
```

Router R2 OSPF Configuration

```
router ospf 1
 network 192.168.20.20 0.0.0.0 area 2
```

```
 network 192.168.85.0 0.0.255.255 area 2
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 2
```

Configuring IS-IS for Interarea Tunnels: Example

The following configuration fragments illustrate how to configure IS-IS for interarea tunnels assuming that:

- R1 and Rx are level-1 routers
- Ra, Ry, and Rb are level-1-2 routers
- Rz and R2 are level-1 routers

Router R1 IS-IS Configuration

```
clns routing
interface POS1/0
 ip router isis
router isis
 metric-style wide
 net 49.0000.0100.0000.0010
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1
```

Router Rx IS-IS Configuration

```
clns routing
interface POS1/0
 ip router isis
interface POS2/0
 ip router isis
router isis
 metric-style wide
 net 49.0000.2000.0100.0001
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1
```

Router Ra IS-IS Configuration

```
clns routing
interface POS2/0
 ip router isis
interface POS3/0
 ip router isis
router isis
 metric-style wide
 net 49.0000.2000.0200.0002
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1
 mpls traffic-eng level-2
```

Router Ry IS-IS Configuration

```
clns routing
interface POS3/0
 ip router isis
interface POS4/0
 ip router isis
router isis
 metric-style wide
 net 49.0000.2000.0300.0003
```

```
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1
 mpls traffic-eng level-2
```

Router Rb IS-IS Configuration

```
clns routing
interface POS4/0
 ip router isis
interface POS5/0
 ip router isis
router isis
 metric-style wide
 net 49.0000.2000.0400.0004
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1
 mpls traffic-eng level-2
```

Router Rz IS-IS Configuration

```
clns routing
interface POS5/0
 ip router isis
interface POS6/0
 ip router isis
router isis
 metric-style wide
 net 49.0000.2000.0500.0005
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1
```

Router R2 IS-IS Configuration

```
clns routing
interface POS6/0
 ip router isis
router isis
 metric-style wide
 net 49.0000.0200.0000.0020
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1
```

Configuring MPLS and RSVP to Support Traffic Engineering: Example

The following configuration fragments show how to configure MPLS and RSVP to support traffic engineering on the routers.

Router R1 Traffic Engineering Configuration

```
ip cef
mpls traffic-eng tunnels
interface Loopback0
 ip address 192.168.10.10 255.255.255.255
interface POS1/0!Each interface supporting MPLS TE must include the following:
 mpls traffic-eng tunnels
 ip rsvp bandwidth 1158
```

The configuration of routers Rx, Ra, Ry, Rb, Rz, and R2 for traffic engineering operation is similar to that for R1.

Configuring an Interarea Tunnel: Example

The following configuration fragments show how to configure a traffic engineering interarea tunnel. Router R1 is the headend for Tunnel1, and Router R2 is the tailend. Tunnel1 is configured with a path option that is loosely routed through Ra and Rb.

R1 Interarea Tunnel Configuration

The following commands configure an MPLS TE tunnel to use explicit paths:

```
interface Tunnel1
 ip unnumbered Loopback0
 tunnel destination 192.168.20.20
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng bandwidth 300
 tunnel mpls traffic-eng path-option 1 explicit name path-tunnel1
```

The following commands configure an explicit path:

```
ip explicit-path name path-tunnel1
 next-address loose 192.168.40.40
 next-address loose 192.168.60.60
 next-address loose 192.168.20.20  !Specifying the tunnel tailend in the loosely routed
!path is optional.
```

Note Generally for an interarea tunnel you should configure multiple loosely routed path options that specify different combinations of ABRs (for OSPF) or level-1-2 boundary routers (for IS-IS) to increase the likelihood that the tunnel will be successfully signaled. In this simple topology there are no other loosely routed paths.

Additional References

The following sections provide references related to the MPLS Traffic Engineering: Interarea Tunnels feature.

Related Documents

Related Topic	Document Title
IS-IS	• *Cisco IOS IP Configuration Guide*, Release 12.2 • *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols*, Release 12.2
Link protection	*MPLS Traffic Engineering Fast Reroute—Link Protection*, Release 12.0(16)ST
MPLS traffic engineering	• *Cisco IOS Switching Services Command Reference*, Release 12.4T • *Cisco IOS Switching Services Configuration Guide*, Release 12.2 • *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols*, Release 12.2
OSPF	• *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols*, Release 12.2 • *Cisco IOS IP Configuration Guide*, Release 12.2

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	—

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and technical documentation. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Command Reference

This section documents only commands that are new or modified.

- **next-address**

next-address

To specify the next IP address in the explicit path, use the **next-address** command in IP explicit path configuration mode. To remove the specified next IP address in the explicit path, use the **no** form of this command.

next-address [**loose** | **strict**] *A.B.C.D*

no next-address *A.B.C.D*

Syntax Description

A.B.C.D	Next IP address in the explicit path.
loose	Specifies that the previous address (if any) in the explicit path need not be directly connected to A.B.C.D, and that the router is free to determine the path from the previous address (if any) to A.B.C.D.
strict	Specifies that the previous address (if any) in the explicit path must be directly connected to A.B.C.D.

Defaults

The next IP address in the explicit path is not specified.

Command Modes

IP explicit path configuration

Command History

Release	Modification
12.0(5)S	This command was introduced.
12.0(19)ST1	The **loose** and **strict** keywords were added.
12.0(21)ST	Support for the Cisco 12000 series router was added.
12.2(18)S	The command was integrated into Cisco IOS Release 12.2(18)S.
12.2(18)SXD	This command was integrated into Cisco IOS Release 12.2(18)SXD.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

To specify an explicit path that includes only the addresses specified, specify each address in sequence by using the **next-address** command without the **loose** keyword.

To configure an interarea traffic engineering (TE) tunnel, configure the tunnel path options as loose explicit paths. Specify that each autonomous system border router (ASBR) traversed by the tunnel LSP is a loose hop by entering the **loose** keyword with the **next-address** command.

To use explicit paths for TE tunnels within an IGP area, you can specify a combination of both loose and strict hops.

Examples

The following example shows how to assign the number 60 to the IP explicit path, enable the path, and specify 10.3.27.3 as the next IP address in the list of IP addresses:

```
Router(config)# ip explicit-path identifier 60 enable
Router(cfg-ip-expl-path)# next-address 10.3.27.3

Explicit Path identifier 60:
    1: next-address 10.3.27.3
```

The following example shows a loose IP explicit path with ID 60. An interarea TE tunnel has a destination of 10.3.29.3 and traverses ASBRs 10.3.27.3 and 10.3.28.3.

```
Router(config)# ip explicit-path identifier 60
Router(cfg-ip-expl-path)# next-address loose 10.3.27.3
Router(cfg-ip-expl-path)# next-address loose 10.3.28.3
Router(cfg-ip-expl-path)# next-address loose 10.3.29.3
```

Related Commands

Command	Description
append-after	Inserts the new path entry after the specified index number.
index	Inserts or modifies a path entry at a specified index.
ip explicit-path	Enters the subcommand mode for IP explicit paths and creates or modifies the specified path.
list	Displays all or part of the explicit paths.
show ip explicit-paths	Displays configured IP explicit paths.

Feature History for MPLS Traffic Engineering: Interarea Tunnels

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/cfn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release. Unless noted otherwise, subsequent releases of that Cisco IOS software release also support that feature.

Table 1 Feature Information for MPLS Traffic Engineering: Interarea Tunnels

Feature Name	Releases	Feature Information
MPLS Traffic Engineering: Interarea Tunnels	12.0(19)ST1 12.0(21)ST 12.2(18)S 12.2(18)SXD 12.2(27)SBC 12.2(28)SB 12.2(33)SRB	The MPLS Traffic Engineering: Interarea Tunnels feature allows you to establish MPLS TE tunnels that span multiple IGP areas and levels, removing the restriction that had required the tunnel headend and tailend routers both be in the same area. 12.0(19)ST1, this feature was introduced. In 12.0(21)ST, support was added for the Cisco 10000 series routers. In 12.2(18)S, this feature was integrated. In 12.2(18)SXD, this feature was integrated. In 12.2(27)SBC, this feature was integrated. In 12.2(28)SB, this feature was integrated. In 12.2(33)SRB, support was added for SSO recovery of LSPs that include loose hops.

Glossary

ABR—Area Border Router. A router connecting two areas. In OSPF, ABRs belong to both areas and must maintain separate topological databases for each. When an OSPF router has interfaces in more than one area, it is an Area Border Router.

area—A logical set of network segments (for example, one that is OSPF-based) and their attached devices. Areas usually are connected to other areas by routers, making up a single autonomous system. OSPF and IS-IS define their areas differently. OSPF area borders are marked by routers. Some interfaces are in one area, and other interfaces are in another area. With IS-IS, all the routers are completely within an area, and the area borders are on links, not on routers. The routers that connect the areas are level-2 routers, and routers that have no direct connectivity to another area are level-1 routers.

area ID—In an IS-IS router, this area address is associated with the entire router rather than an interface. A router can have up to three area addresses. Both the area ID and the system ID are defined on an IS-IS router by a single address, the Network Entry Title (NET).

autonomous system—A collection of networks under a common administration sharing a common routing strategy. Autonomous systems are subdivided by areas.

Cisco Express Forwarding—An advanced Layer 3 IP switching technology. Cisco Express Forwarding optimizes network performance and scalability for networks that have large and dynamic traffic patterns, such as the Internet, and for networks characterized by intensive Web-based applications or interactive sessions. Cisco Express Forwarding uses a Forwarding Information Base (FIB) to make IP destination prefix-based switching decisions. The FIB is conceptually similar to a routing table or information base. When routing or topology changes occur in the network, the IP routing table is updated, and those changes are reflected in the FIB. The FIB maintains next-hop address information based on the information in the IP routing table.

headend—The upstream, transmit end of a tunnel. The router that originates and maintains the traffic engineering LSP.

IGP—Interior Gateway Protocol. Internet protocol used to exchange routing information within an autonomous system. Examples of common IGPs include OSPF and Routing Information Protocol (RIP).

interarea TE—Ability for a traffic engineering LSP to span multiple areas.

IS-IS—Intermediate System-to-Intermediate System. IS-IS is an OSI link-state hierarchial routing protocol based on DECnet Phase V routing, where intermediate system (IS) routers exchange routing information is based on a single metric to determine the network topology.

label-switched path (LSP) tunnel—A configured connection between two routers in which label switching is used to carry the packets.

level-1 routers—Routers that are directly connected to other areas. The routers are not in the backbone. MPLS does not run in the background. These routers are also called internal routers.

level-2 routers—Routers that connect two areas. These routers let you run MPLS in the background.

load balancing—The distribution of traffic among multiple paths to the same destination so that the router uses bandwidth efficiently. Load balancing increases the use of network segments, thus increasing effective network bandwidth.

LSP—label-switched path. A sequence of hops (such as R0...Rn) in which a packet travels from R0 to Rn through label switching mechanisms. A label-switched path can be chosen dynamically, based on normal routing mechanisms, or through configuration.

mask—A bit combination used to describe which part of an address refers to the network or the subnet and which part refers to the host.

MPLS—Multiprotocol Label Switching. A method for forwarding packets (frames) through a network. It enables routers at the edge of a network to apply labels to packets. ATM switches or existing routers in the network core can switch packets according to the labels with minimal lookup overhead.

OSPF—Open Shortest Path First. Link-state, hierarchical IGP routing algorithm proposed as a successor to Routing Information Protocol (RIP) in the Internet community. OSPF features include least-cost routing, multipath routing, and load balancing.

process ID—Distinguishes one process from another within the device. An OSPF process ID can be any positive integer, and it has no significance outside the router on which it is configured.

router ID—Something by which a router originating a packet can be uniquely distinguished from all other routers. For example, an IP address from one of the router's interfaces.

tailend—The downstream, receive end of a tunnel. The router that terminates the traffic engineering LSP.

traffic engineering—The techniques and processes that cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods were used.

tunnel—A secure communication path between two peers, such as two routers. A traffic engineering tunnel is a label-switched tunnel that is used for traffic engineering. Such a tunnel is set up through means other than normal Layer 3 routing; it is used to direct traffic over a path different from the one that Layer 3 routing could cause the tunnel to take.

virtual link—Ordinarily, each area is directly connected to area 0. A virtual link is used for a connection when an area is connected to an area that is one area away from area 0.

Note See *Internetworking Terms and Acronyms* for terms not included in this glossary.